View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Elsevier - Publisher Connector

ELSEVIER

Theoretical Computer Science 218 (1999) 325–346

Theoretical Computer Science

Two-dimensional cellular automata recognizer

Véronique Terrier *

GREYC, Université de Caen, Campus 2, 14032 Caen, France

Abstract

We are investigating cellular automata on two-dimensional array as language recognizer. Linear acceleration for Moore and Von Neumann neighborhood is presented. Relationships with one-dimensional CAs and Turing machines are considered. Some limitations of the power capabilities of real-time recognition are shown. © 1999 Published by Elsevier Science B.V. All rights reserved.

1. Introduction

Cellular automata appear to be a relevant model for massively parallel computation. To evaluate their computation capability a lot of interest has been devoted to one-dimensional CAs as language recognizer. In recent years several papers investigate various types of two-dimensional CAs [1, 4, 5, 8]. Here we will investigate some basic properties of two-dimensional CAs as two-dimensional language recognizer with parallel input mode and bounded computation. Indeed in a practical point of view two-dimensional array looks more natural. Clearly on two-dimensional array the behavior of CAs becomes more complex. For instance with regard to the global properties the reversibility and surjectivity problems become indecidable in dimension two [6]. About the language recognition when the input mode is sequential Cole [2] has shown that the power strictly increases with the dimension of the space. But actually to solve problems in higher dimensions often consists in extending methods developed for the dimension one. Here we will generalize techniques used in one-dimensional CAs.

There are several possible definitions of two-dimensional CA recognizers according to the choice of the neighborhood, the different ways to supply the input and to get the result. In this paper we will restrict to Moore and Von Neumann neighborhood, with a parallel input mode where the inputs are supplied in an array rectangular in shape and with the result of the computation is get on a distinguished cell (more often the upper-leftmost cell). The definitions are specified in Section 2. In Section 3 we adapt the

* E-mail: veroniqu@info.unicaen.fr.

0304-3975/99/$ - see front matter © 1999 Published by Elsevier Science B.V. All rights reserved.
PII: S0304-3975(98)00329-6

linear speed-up methods known for one-dimensional CAs to two-dimensional CAs. In particular for Von Neumann neighborhood this generalization leads to a solution where the grouped cells overlap each other. In Sections 4 and 5 we compare two-dimensional CAs with one-dimensional CAs and Turing machines. The last section focuses on real-time computation and its limitation using counting arguments developed before for one-dimensional CAs. In particular we present a language not real-time recognizable with Moore neighborhood which is real-time recognizable with Von Neumann neighborhood.

2. Definitions

A two-dimensional cellular automaton is a two-dimensional array of identical finite automata (cells) indexed by Z^2 and working synchronously. Each cell evolves according its neighborhood at discrete time step. Formally a two-dimensional CA is a 3-tuple (S,N,δ) where S is the set of states, $N=\{(x_1,y_1),\ldots,(x_k,y_k)\}\subset Z^2$ the neighborhood, $\delta: S^k \to S$ the transition function. Denoting $\langle (u,v),t\rangle$ the state of the cell (u,v) at time t, we have $\langle (u,v),t\rangle = \delta(\langle (u+x_1,v+y_1),t-1\rangle,\ldots,\langle (u+x_k,v+x_k),t-1\rangle)$.

The two classic neighborhoods are considered in this paper: the Moore neighborhood where $N=\{(x,y):\ \|(x,y)\|_\infty \leqslant 1\}$, the Von Neumann neighborhood where $N=\{(x,y):\ \|(x,y)\|_1 = 1\}$.

Here we are interested in CAs as language recognizer. For that purpose we distinguish two subsets of S : Σ and S_{accept}; Σ corresponds to the alphabet of the language and S_{accept} is the subset of accepting states. We specify also a distinguished cell, more often the cell (1,1), which communicates with outside.

In a natural way the two-dimensional languages will be investigated but also to compare two-dimensional CAs with other one-dimensional models the one-dimensional languages will be examined. For two-dimensional languages we need the classic definitions developed for two-dimensional objects. A picture over an alphabet Σ is a $m*n$ array of elements of Σ. The size of the picture is denoted by (m,n). $p(i,j)$, where $1\leqslant i\leqslant m$ and $1\leqslant j\leqslant n$, denotes the element on line i and column j of the picture p. Σ^{**} denotes the set of all pictures over Σ and Σ^{m*n} the set of all pictures of size (m,n).

We will consider only parallel input mode: at initial time 0, for $1\leqslant i\leqslant n$ and $1\leqslant j\leqslant m$, each element $p(i,j)$ of the input picture p is fed to the cell (i,j). We restrict to bounded computation, so the other cells remain in a special state # during all the computation. We say that a CA accepts the picture p in t steps if the distinguished cell enters an accepting state at time t. Let T be a function from N^2 to N. We say that a CA recognizes the language L in time T if it accepts the pictures p of size (m,n) in time $T(m,n)$. As in one-dimensional case we define real time as the minimal time needed by the distinguished cell to read any particular part of the input. Thus when the distinguished cell is the cell (1,1), the real time corresponds to the function $T(m,n)=\max(m,n)-1$ for Moore neighborhood and the function $T(m,n)=m+n-2$ for Von Neumann neighborhood. And for a constant $c>1$, $T(m,n)=c\max(m,n)$ (resp. $c(m+n)$) gives rise to linear time. $\text{CA}_{\text{Moore}}(T(m,n))$ (resp. $\text{CA}_{\text{VN}}(T(m,n))$) will denote the class of

2-dimensional languages recognized in time $T(m,n)$ by a CA with Moore (resp. Von Neumann) neighborhood.

For one-dimensional language recognition by two-dimensional CAs the first ambiguity is in the way the input words are fed into two-dimensional array. For our practical purposes we will consider the words are written in a square in a snakelike order manner. So a word of length n will correspond to a picture of size $(\lceil\sqrt{n}\rceil, \lceil\sqrt{n}\rceil)$ with the letters successively written from left to right and from right to left, and where the last line is completed with $\lceil\sqrt{n}\rceil^2 - n$ blank symbols β.

For example $a_1 \ldots a_{14}$ corresponds to the picture

$$\begin{array}{cccc} a_1 & a_2 & a_3 & a_4 \\ a_8 & a_7 & a_6 & a_5 \\ a_9 & a_{10} & a_{11} & a_{12} \\ \beta & \beta & a_{14} & a_{13} \end{array}$$

Let f be a function from N to N. We say that a CA recognizes the language L in time f if it accepts the words w of length n in time $f(n)$. Thus when the distinguished cell is the cell (1,1), the real time corresponds to the function $f(n) = \lceil\sqrt{n}\rceil - 1$ for Moore neighborhood and the function $f(n) = 2\lceil\sqrt{n}\rceil - 2$ for Von Neumann neighborhood. $\mathrm{CA}_{\mathrm{Moore}}(f(n))$ (resp. $\mathrm{CA}_{\mathrm{VN}}(f(n))$) will denote the class of 1-dimensional languages recognized in time $f(n)$ by a CA with Moore (resp. Von Neumann) neighborhood.

3. Linear acceleration

The linear acceleration known for one-dimensional CAs (cf. [3, 7]) which establishes that a language recognized in time $f(n)$ is also recognized in time $n + \lceil(f(n)-n)/k\rceil$ for some k could be generalized to two-dimensional CAs. An initial phase (of $\max(m,n)$ steps for Moore neighborhood and $m+n-1$ steps for Von Neumann neighborhood) will consist in compacting the cells by group of $k \times k$ cells. After this phase the CA could work k times faster for Moore neighborhood and $(k+1)/2$ times faster for Von Neumann neighborhood. So a language recognized in time $T(m,n)$ will be also recognized in time $\max(m,n) + \lceil(T(m,n) - \max(m,n))/k\rceil$ with Moore neighborhood and in time $m+n-1+2\lceil(T(m,n)-m-n+1)/(k+1)\rceil$ with Von Neumann neighborhood.

3.1. The Moore neighborhood case

Let A be a CA which recognizes the language L in time $T(m,n)$. We will build a CA B which recognizes the language L in time $\max(m,n) + \lceil(T(m,n) - \max(m,n))/k\rceil$ for some k. The features of the CA B are as described in [7] for the one-dimensional case: while computing at the same speed as the initial CA A, the CA B performs grouping and synchronization to stop it and launch an automaton which runs k times faster than the initial CA A. The grouping process is performed both on the lines and the columns. The lines grouping process propagates leftwards and is such that at time

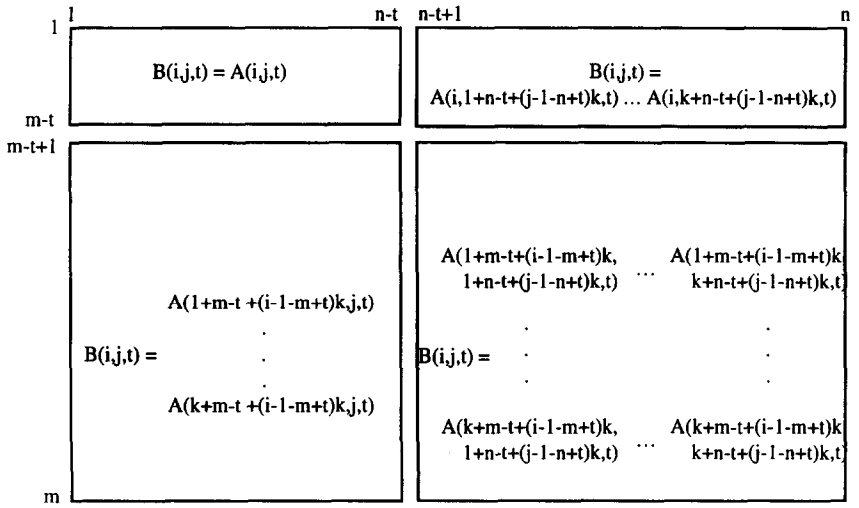

Fig. 1. The grouping process at time t.

t the lines $m-t+1, m-t+2, \ldots, m$ are grouped by k; so to stop it a synchronisation is required at time m. Symmetrically the columns grouping process propagates upwards and must be stopped at time n.

We first describe the synchronization processes. Recall that it exists a one- dimensional CA which synchronizes a segment of length n in $n-1$ steps provided the two extremities be set initially in special states G_{left} and G_{right} (see [7]). So on the automaton B we synchronize each column in order to synchronize all cells at time m and each line in order to synchronize all cells at time n. For that purpose on each cell (i,j) one state is devoted to the synchronization of the line i and a second one to the synchronization of the column j. At time 1 for each cell these two states are set in the special states G_{left} or G_{right} or in a quiescent state according their west, east, north and south neighbors are in the border state #.

We now describe the grouping process. See Fig. 1. We need the following notations. $A(i,j,t)$ will denote the state of the cell (i,j) at time t on the initial CA A (in particular $A(i,j,t)$ is the border state # if $i>m$ or $j>n$); $B(i,j,t)$ will denote the state of the cell (i,j) at time t on the CA B.

$$L_{\text{left}}(i,t) \text{ will represent } \begin{cases} i & \text{if } i \leqslant m-t \text{ and } t \leqslant m, \\ 1+m-t+(i-(1+m-t))k & \text{if } i > m-t \text{ and } t \leqslant m, \\ 1+(i-1)k & \text{if } t > m, \end{cases}$$

$$C_{\text{top}}(j,t) \begin{cases} j & \text{if } j \leqslant n-t \text{ and } t \leqslant n, \\ 1+n-t+(j-(1+n-t))k & \text{if } j > n-t \text{ and } t \leqslant n, \\ 1+(j-1)k & \text{if } t > n, \end{cases}$$

$$L_{\text{right}}(i,t) \begin{cases} L_{\text{left}}(i,t) & \text{if } i \leqslant m-t, \\ L_{\text{left}}(i,t)+k-1 & \text{if } i > m-t, \end{cases}$$

and

$$C_{\text{bottom}}(j,t)\begin{cases} C_{\text{top}}(j,t) & \text{if } j\leqslant n-t,\\ C_{\text{top}}(j,t)+k-1 & \text{if } j>n-t.\end{cases}$$

Proposition 1. *For any CA A there exists a CA B such that if at time* 0 *the two CA A and B are in the same configuration then at time t with* $t\leqslant\max(m,n)$, $B(i,j,t)$ *contains the following array of states*:

$$\begin{matrix} A(L_{\text{left}}(i,t),C_{\text{top}}(j,t),t) & \cdots & A(L_{\text{right}}(i,t),C_{\text{top}}(j,t),t)\\ \vdots & \ddots & \vdots\\ A(L_{\text{left}}(i,t),C_{\text{bottom}}(j,t),t) & \cdots & A(L_{\text{right}}(i,t),C_{\text{bottom}}(j,t),t)\end{matrix};$$

and from time $t>\max(m,n)$ *the CA B can perform in one step k steps of the CA A.*

Proof. At time 1, $L_{\text{left}}(i,1)$ is i, $L_{\text{right}}(i,1)$ is i for $i<m$ and $i+k-1$ for $i=m$, $C_{\text{top}}(j,1)$ is j, $C_{\text{bottom}}(j,1)$ is j for $j<n$ and $j+k-1$ for $j=m$. Hence we could define a transition function such that the site (m,n) whose east and south neighbors are in state # enters in state $\begin{matrix} A(m,n,1) & \#\cdots\# \\ \# & \# \quad \# \\ \vdots & \ddots\ \vdots \\ \# & \cdots\ \# \end{matrix}$; the rightmost sites (i,n), with $i<m$, whose east neighbors are in state # enters in state $A(i,n,1)\#\ldots\#$; the lowest sites (m,j), with $j<n$, whose north neighbors are in state # enters in state $\begin{matrix} A(m,j,1)\\ \#\\ \vdots\\ \#\end{matrix}$; the other sites evolve as on the CA A.

At times $t\leqslant\min(m,n)$, observe that as $L_{\text{left}}(i,t)-1\geqslant L_{\text{left}}(i-1,t-1)$, $L_{\text{right}}(i,t)+1\leqslant L_{\text{right}}(i+1,t-1)$, $C_{\text{top}}(j,t)-1\geqslant C_{\text{top}}(j-1,t-1)$, $C_{\text{bottom}}(j,t)+1\leqslant C_{\text{bottom}}(j+1,t-1)$, we have the required information to compute all the states $A(u,v,t)$ with $L_{\text{right}}(i,t)\leqslant u\leqslant L_{\text{left}}(i,t)$ and $C_{\text{bottom}}(j,t)\leqslant v\leqslant C_{\text{top}}(j,t)$ on the site (i,j) at time t. And as $L_{\text{right}}(i,t)=L_{\text{right}}(i+1,t-1)-1$, $L_{\text{left}}(i,t)=L_{\text{left}}(i+1,t-1)-1$, $C_{\text{bottom}}(j,t)=C_{\text{bottom}}(j+1,t-1)-1$, $C_{\text{top}}(j,t)=C_{\text{top}}(j+1,t-1)-1$, we are able to localize the relevant states $A(u,v,t)$ with $L_{\text{left}}(i,t)\leqslant u\leqslant L_{\text{right}}(i,t)$ and $C_{\text{top}}(j,t)\leqslant v\leqslant C_{\text{bottom}}(j,t)$ among all the states $A(u,v,t)$ with $L_{\text{left}}(i-1,t-1)+1\leqslant u\leqslant L_{\text{right}}(i+1,t-1)-1$ and $C_{\text{top}}(j-1,t-1)+1\leqslant v\leqslant C_{\text{bottom}}(j+1,t-1)-1$ we may compute.

At times t with $\min(m,n)<t\leqslant\max(m,n)$, for symmetry reasons we can suppose without loss of generality that $m\leqslant n$. Observe that as $L_{\text{left}}(i,t)-1\geqslant L_{\text{left}}(i-1,t-1)$, $L_{\text{right}}(i,t)+1\leqslant L_{\text{right}}(i+1,t-1)$, $C_{\text{top}}(j,t)-1\geqslant C_{\text{top}}(j-1,t-1)$, $C_{\text{bottom}}(j,t)+1\leqslant C_{\text{bottom}}(j+1,t-1)$, we have the required information to compute all the states $A(u,v,t)$ with $L_{\text{left}}(i,t)\leqslant u\leqslant L_{\text{right}}(i,t)$ and $C_{\text{top}}(j,t)\leqslant v\leqslant C_{\text{bottom}}(j,t)$ on the site (i,j) at time t. And as $L_{\text{right}}(i,t)=L_{\text{right}}(i,t-1)-1$, $L_{\text{left}}(i,t)=L_{\text{left}}(i,t-1)-1$, $C_{\text{bottom}}(j,t)=$

$C_{\text{bottom}}(j+1,t-1)-1$, $C_{\text{top}}(j,t)=C_{\text{top}}(j+1,t-1)-1$, we are able to determine, provided the synchronization at time m, the new localization of the relevant states.

From the time $t>\max(m,n), L_{\text{left}}(i,t)-k=L_{\text{left}}(i-1,t-1)$, $L_{\text{right}}(i,t)+k=L_{\text{right}}(i+1,t-1)$, $C_{\text{top}}(j,t)-k=C_{\text{top}}(j-1,t-1)$, $C_{\text{bottom}}(j,t)+k=C_{\text{bottom}}(j+1,t-1)$. Thus we have the required information to compute k steps of the CA A in one step on the CA B. And the synchronization at time $\max(m,n)$ allows B to enter a new behavior.

3.2. *The Von Neumann neighborhood case*

See Figs. 2–5. For Von Neumann neighborhood the grouping process is performed by compacting not the lines and the columns but the diagonals. Precisely three grouping processes will interact: from the cell $(m,1)$ toward the diagonal $i=j$, the diagonals $i-j=m-t$ for $t=1,\ldots,m$ will be grouped by k in m steps; from the cell $(1,n)$ toward the diagonal $i=j$, the diagonals $j-i=n-t$ for $t=1,\ldots,n$ will be grouped by k in n steps; from the cell (m,n) toward the diagonal $i+j=2$, the diagonals $i+j=m+n-t+1$ for $t=1,\ldots,m+n-1$ will be grouped by k in $m+n-1$ steps. In order to compute at the same speed than the initial CA A, the grouping CA B will pack together only cells of same parity (cells (i,j) with $i+j$ of same parity). Actually the grouped cells will overlap. We choose also k odd.

To stop these grouping processes, synchronizations will be required at times m,n and $m+n-1$. For that purpose we synchronize each column in order to synchronize all cells at time m, each line in order to synchronize all cells at time n and in addition from time m we synchronize each line in order to synchronize all cells at time $m+n-1$.

We will now focus on the grouping process of the diagonals $i-j=m-t$ for $t=1,\ldots,m$ without dealing with the other grouping processes. This grouping process is initiated from the cell $(m,1)$, propagates to the Northeast reaching at time t the diagonal $i-j=m-t$ and stops at time m on the diagonal $i=j$. At time t the grouping process works only on the diagonals $i-j=\alpha$ with α of same parity than $m-t$ in shifting $k-1$ diagonals of each k-grouped diagonals to the next one of same parity. To describe more precisely the process we need the following notations. We denote by $A(i,j,t)$ the state of the cell (i,j) at time t on the initial CA A; we denote by $B(i,j,t)$ the state of the cell (i,j) at time t on the grouping CA B. $G(i,j,t)$ denotes $A(i,j,t)$ if $i-j<m-t$ else the sequence of the k sites $A(u,v,t)$ such that $u+v=i+j$ and $i-j+2\lfloor\frac{1}{2}[i-j-(m-t)]\rfloor(k-1)\leqslant u-v\leqslant i-j+2\lfloor\frac{1}{2}[i-j-(m-t)]\rfloor(k-1)+2(k-1)$; $H(i,j,t)$ denotes the sequence of the k sites $A(u,v,t)$ such that $u+v=i+j$ and $i-j+2\lfloor\frac{1}{2}[i-j-(m-(t+1))]\rfloor(k-1)\leqslant u-v\leqslant i-j+2\lfloor\frac{1}{2}[i-j-(m-(t+1))]\rfloor(k-1)+2(k-1)$.

Proposition 2. *For any CA A there exists a CA B such that if at time 0 the two CAs A and B are in the same configuration then at times $t<m$, $B(i,j,t)$ contains $A(i,j,t)$ if $i-j<m-t$, $G(i,j,t)$ if $i-j\geqslant m-t$ and $i+j+m+t$ is odd, $G(i,j,t)$, $G(i+1,j,t-1)$, $H(i,j,t-1)$ if $i-j\geqslant m-t$ and $i+j+m+t$ is even and from time $t\geqslant m$, $B(i,j,t)$ contains the sites $A(u,v,t)$ with $u+v=i+j$, $\max(0,(i-j)k-(k-1))\leqslant u-v\leqslant(i-j)k+k-1$.*

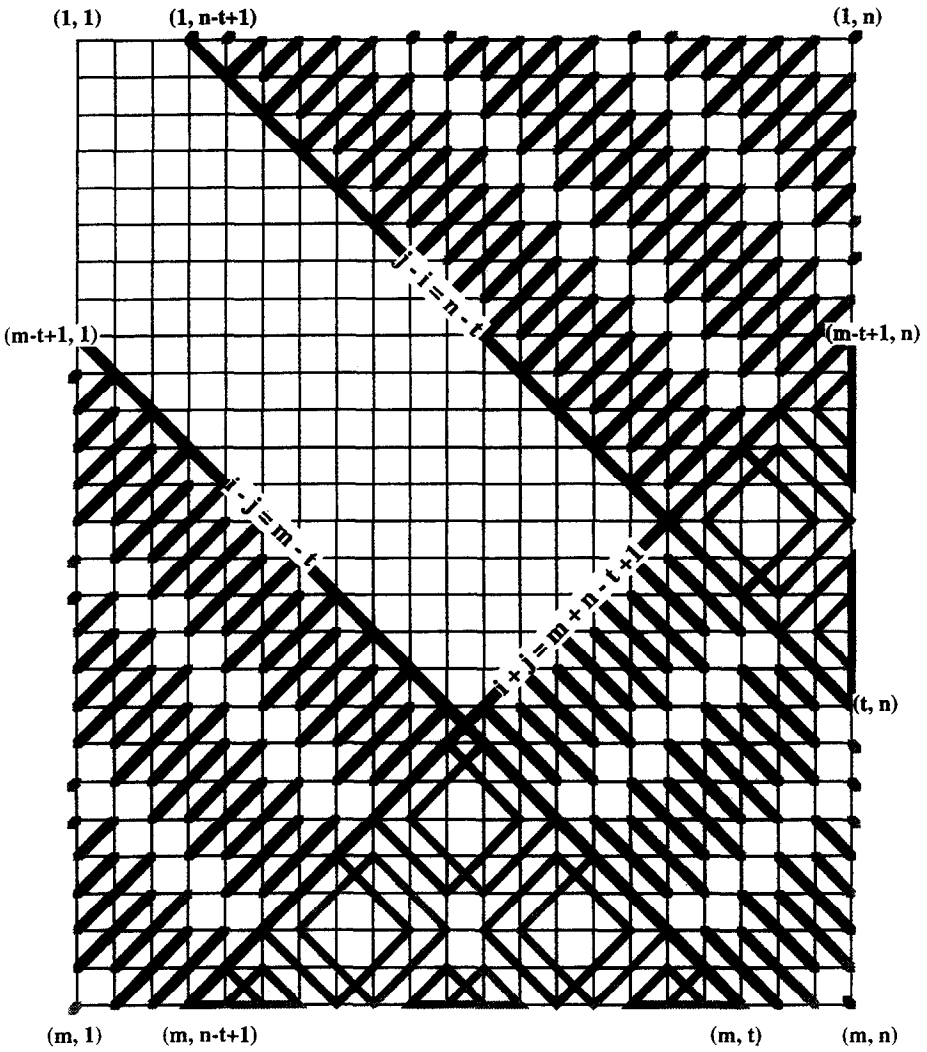

Fig. 2. Time $t < \min(n, m)$.

Proof. At time 1 on the site $(m, 1)$ whose both west and south neighbors are in border state # a wave is initiated which spreads at maximal speed to the north and the east. So this wave proceeds at time t on the cells (i, j) such that $i - j \geqslant m - t$ and is able to discriminate the parity of $i + j + m + t$. Then in the case of $i - j < m - t$ the behavior of the CA B is the same one of the CA A. In the case of $i - j \geqslant m - t + 1$ and $i + j + m + t + 1$ odd, note that $G(i, j, t + 1)$ deals with the same cells as $G(i, j, t)$. And clearly $G(u, v, t)$ with $(u - i, v - j) \leqslant 1$ contain all the required states to compute $G(i, j, t + 1)$. In the case of $i - j \geqslant m - t + 1$ and $i + j + m + t + 1$ even, from $H(i, j - 1, t - 1), G(i + 1, j - 1, t - 1)$ and $H(i + 1, j, t - 1)$ we can compute on the cell (i, j) at time $t + 1$ the sites $A(u, v, t)$ with $u + v = i + j$ and $i - j + 2 + 2\lfloor \frac{1}{2}(i - j - (m - t)) \rfloor (k - 1) \leqslant u - v \leqslant i - j + 2 +$

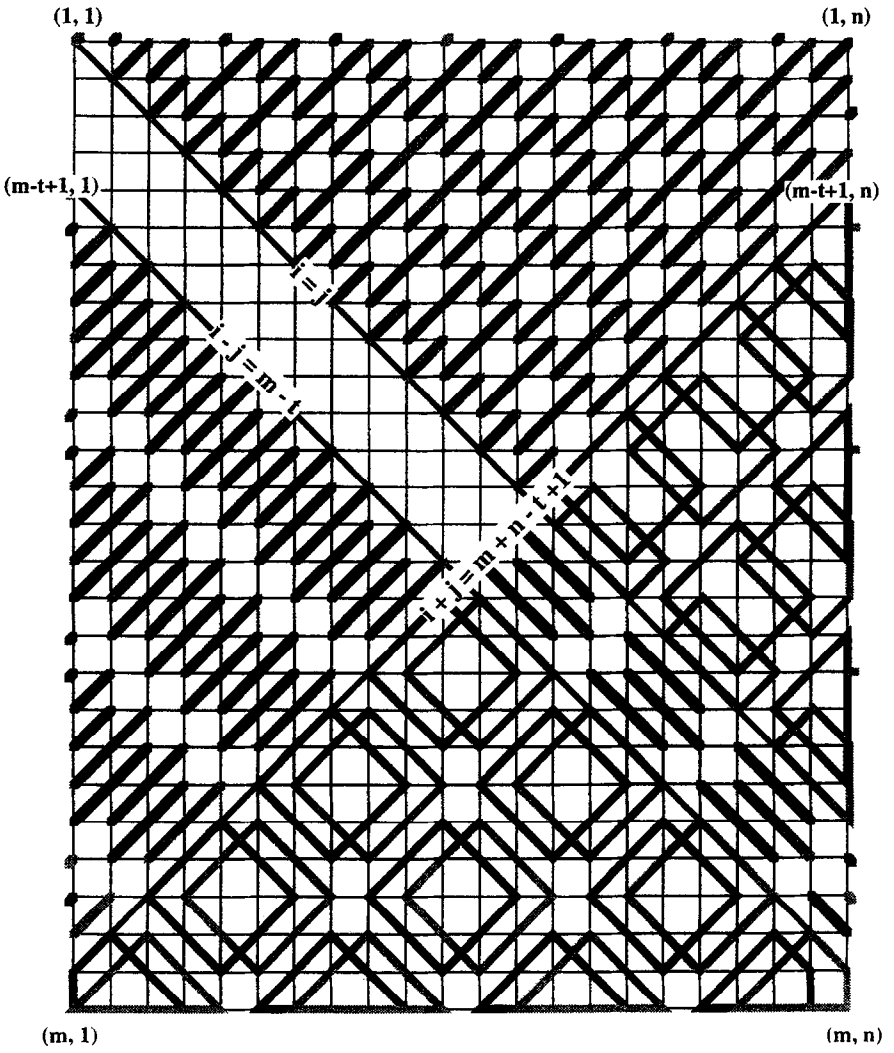

Fig. 3. $\min(n,m) \leqslant \text{time } t < \max(m,n)$.

$2\lfloor\frac{1}{2}(i-j-(m-t))\rfloor(k-1)+2(k-1)$; in addition $G(i,j,t)$ contains the state of the site $A(u,v,t)$ with $u+v=i+j$ and $u-v=i-j+2\lfloor\frac{1}{2}(i-j-(m-t))\rfloor(k-1)$. So we get $H(i,j,t)$. Moreover from $H(i,j,t)$ and $G(u,v,t)$ with $(u-i,v-j)\leqslant 1$ we have all the required states to compute $G(i,j,t+1)$. Besides $G(i+1,j,t-1)$ comes from the south neighbor.

At time m when the synchonization occurs, the process works as before on the cell (i,j) of odd parity. But on the cells (i,j) of even parity which contain k sites of the CA A, only $(k-1)/2$ (instead $k-1$ as before) sites of the CA A are shifted to the even cells $(i-1,j+1)$. Therefore $B(i,j,m)$ contains the sites $A(u,v,m)$ with

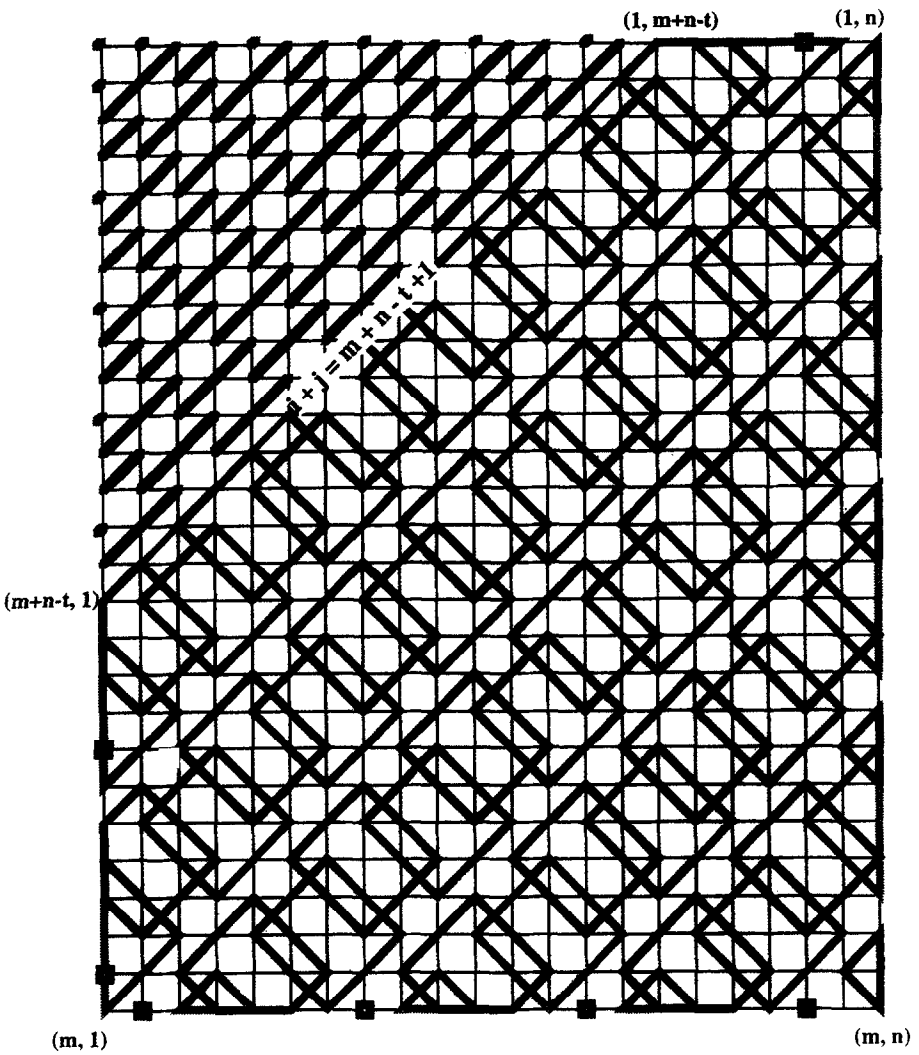

Fig. 4. $\max(m,n) \leqslant \text{time}\, t < m+n-1$.

$u+v=i+j$, $(i-j)k-(k-1) \leqslant u-v \leqslant (i-j)k+k-1$ if $u-v>0$ or $0 \leqslant u-v \leqslant k-1$ if $u=v$. And after the step m the grouping process stops.

Actually the same process is achieved on the diagonals $j-i=n-t$ for $t=1,\ldots,n$ from the cell $(1,n)$ toward the diagonal $i=j$ in n steps and on the diagonals $i+j=m+n-t+1$ for $t=1,\ldots,m+n-1$ from the cell (m,n) toward the diagonal $i+j=2$ (cf. Figs. 2–5). Moreover, a composition (which we do not detail) of the grouping processes of the diagonals $i-j=m-t$ $(t=1,\ldots,m)$ and $i+j=m+n-t+1$ $(t=1,\ldots,m+n-1)$ occurs on the sites $B(i,j,t)$ with $i-j \geqslant m-t$ and $i+j \geqslant m+n-t+1$ at times $t \leqslant m$, and another composition of the grouping processes of the diagonals $j-i=n-t$ $(t=1,\ldots,n)$

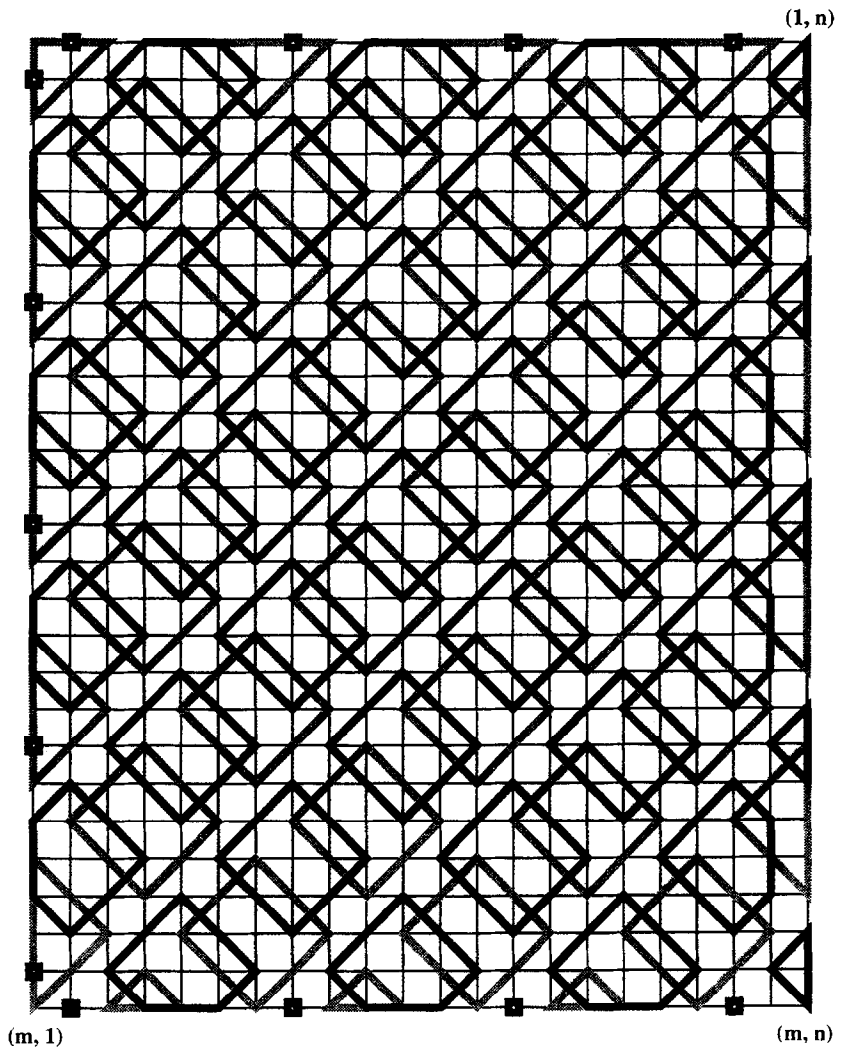

Fig. 5. time $t \geqslant m+n-1$.

and $i+j=m+n-t+1$ $(t=1,\ldots,m+n-1)$ occurs on the sites $B(i,j,t)$ with $j-i \geqslant n-t$ and $i+j \geqslant m+n-t+1$ at times $t \leqslant n$.

Consequently, at time $m+n-1$ $B(i,j,t)$ contains the sites $A(u,v,t)$ such that $u+v$ and $i+j$ have the same parity, $(i-j)k-(k-1) \leqslant u-v \leqslant (i-j)k+k-1$ and $(i+j-2)k-(k-1) \leqslant u+v-2 \leqslant (i+j-2)k+k-1$. In other words $B(i,j,t)$ contains the sequence of k^2 sites $A(1+(i-1)k+\alpha, 1+(j-1)k+\beta, t)$ with $\|(\alpha,\beta)\|_1 \leqslant k-1$ and $\alpha+\beta$ even. Therefore the site $B(i+\alpha, j+\beta, t)$ with $\|(\alpha,\beta)\|_1 \leqslant 2$ contains the sites $A(1+(i-1)k+\alpha, 1+(j-1)k+\beta, t)$ with ($\|(\alpha,\beta)\|_1 \leqslant 3k-1$ and $\alpha+\beta$ even) or ($\|(\alpha,\beta)\|_1 \leqslant 2k-1$ and $\alpha+\beta$ odd) in particular the sites $A(1+(i-1)k+\alpha, 1+(j-1)k+\beta, t)$ with $\|(\alpha,\beta)\|_1 \leqslant 2k$. Hence from this time, $k+1$ steps of the CA A can be

simulated in 2 steps on the CA B. Actually for any integer $s, sk+1$ steps of the CA A can be simulated in $s+1$ steps on the CA B.

Corollary 1. *If* $T(m,n) \geqslant c(m+n)$ *for any* $c>1$, *then* $\mathrm{CA}_{\mathrm{VN}}(T(m,n)) = \mathrm{CA}_{\mathrm{Moore}}(T(m,n))$.

Proof. $\mathrm{CA}_{\mathrm{VN}}(T(m,n) \subset \mathrm{CA}_{\mathrm{Moore}}(T(m,n))$ indeed the Von Neumann neighborhood is contained in the Moore neighborhood.

$\mathrm{CA}_{\mathrm{Moore}}(T(m,n)) \subset \mathrm{CA}_{\mathrm{VN}}(2T(m,n))$ since one step on a Moore CA can be simulated in two steps on a Von Neumann CA. Let k be an odd integer such that $k \geqslant 4c/(c-1)$, it exists since $c>1$.

We have

$$\begin{aligned}\mathrm{CA}_{\mathrm{VN}}(2T(m,n)) &\subset CA_{\mathrm{VN}}(m+n+2)\lceil(2T(m,n)-m-n)/(k+1)\rceil) \\ &\quad \text{according to Section 3.2} \\ &\subset \mathrm{CA}_{\mathrm{VN}}(T(m,n)/c + 4T(m,n)/k) \quad \text{as } T(m,n) \geqslant c(m+n) \\ &\subset \mathrm{CA}_{\mathrm{VN}}(T(m,n)) \quad \text{by choice of } k.\end{aligned}$$

4. Comparison with one-dimensional CAs

Note that the way the input words are supplied into two-dimensional array will be determining in the comparison of one- and two-dimensional CAs. Recall that here we consider the words are written in a square in a snakelike order manner. The following propositions depend closely on this choice.

4.1. Simulation of a one-dimensional CA by a two-dimensional CA

Proposition 3. *If the language L is recognized by a one-dimensional CA in time* $f(n)$ *then L is recognized by a two-dimensional CA with Von Neumann neighborhood in time* $\lceil\sqrt{n}\rceil + f(n)$.

Proof. As the input words are written in a snakelike order, to identify the relevant neighborhood of each site of the square is simple. For that purpose each site will be marked with a state indicating its respective left and right neighbors (NW for the sites with the left neighbor on the north and the right neighbor on the west, and in the same way EW, ES, WS, WE, NE) (Fig. 6).

This marking process propagates downwards one line by step in the following way.

At time 1, the site $(1,1)$ whose north and east neighbors are in state #, enters in state NW; the site $(1,\lceil\sqrt{n}\rceil)$ whose north and west neighbors are in state #, enters in state ES; the sites $(1,i)$ with $1<i<\lceil\sqrt{n}\rceil$ whose north neighbor but not their east or their west neighbors are in state #, enter in state EW.

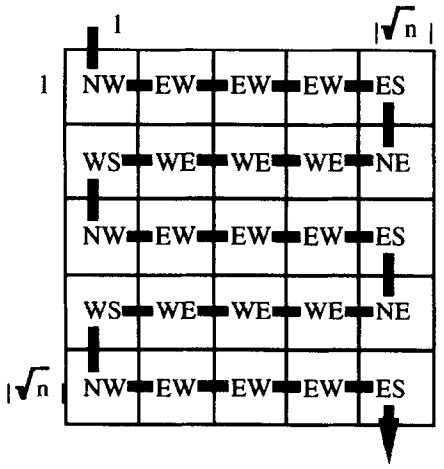

Fig. 6.

At time $i>1$, the sites not marked enter in state WS, WE, NE, NW, EW, ES, respectively if their north neighbor is in state NW, EW, ES, WS, WE, NE, respectively.

Therefore at time $\lceil\sqrt{n}\rceil$ all sites are marked.

Parallely at time 1 a synchronization process starts from both ends of each line. So all cells are synchronized at time $\lceil\sqrt{n}\rceil$. Then from this time the two-dimensional CA operates as in one-dimensional case.

Corollary 2. *If the language L is recognized by a one-dimensional CA in time $f(n)$ then L is recognized by a two-dimensional CA with Von Neumann neighborhood in time $f(n)/k$.*

Proof. Indeed in one-dimensional case the minimal time is n, therefore of greater order than $\lceil\sqrt{n}\rceil$. And according to Section 3.2, we have the desired linear acceleration.

Corollary 3. *If the language L is recognized by a one-dimensional CA in time $f(n)$ then L is recognized by a two-dimensional CA with Moore neighborhood in time $f(n)/k$.*

4.2. Simulation of a two-dimensional CA by a one-dimensional CA

The following proposition is a special case of a result in [1].

Proposition 4. *If the language L is recognized by a two-dimensional CA A (with Moore or Von Neumann neighborhood) in time $f(n)$ then L is recognized by a one-dimensional CA B in time $O(\sqrt{n}f(n))$.*

Proof. On the two-dimensional CA A an input word of length n is supplied in a snakelike order as a picture of size $(\lceil\sqrt{n}\rceil,\lceil\sqrt{n}\rceil)$. So the initial phase of the simulation consists in dividing the working area of the one-dimensional CA B in $\lceil\sqrt{n}\rceil$ segments

of length $\lceil\sqrt{n}\rceil$ so that the ith segment of B corresponds to the ith line of A if i is odd, the reverse of the ith line of A if i is even. For that purpose we distinguish the leftmost cells of each segments i.e. the cells $1+i\lceil\sqrt{n}\rceil$ with $i=0,\ldots,\lceil\sqrt{n}\rceil-1$. Precisely we construct a CA which distinguishes the sites $(1+i\lceil\sqrt{n}\rceil,(i-1)^2+i\lceil\sqrt{n}\rceil)$. The construction is displayed by the Figs. 7 and 8. In parallel a signal end of line is sent from the leftmost cell n at time 0, it meets the signal P on the section corresponding to the diagonal $t=c-1+(\lceil\sqrt{n}\rceil-1)^2$. Therefore among the distinguished sites $(1+i\lceil\sqrt{n}\rceil,(i-1)^2+i\lceil\sqrt{n}\rceil)$ we can select the sites on the diagonal $t=c-1+(\lceil\sqrt{n}\rceil-1)^2$ i.e. the sites $(1+i\lceil\sqrt{n}\rceil,(\lceil\sqrt{n}\rceil-1)^2+i\lceil\sqrt{n}\rceil)$. From these sites vertical signals are initiated, they mark the cells $1+i\lceil\sqrt{n}\rceil$ with $i=0,\ldots,\lceil\sqrt{n}\rceil-1$. During this initial phase each input bit remains on its cell and a synchronization process is initiated on the cell 1 at time 0, in this way the cells are synchronized at time $2n-1$. From this time the simulation of the step k of the CA A will be done at time $2n-1+k(\lceil\sqrt{n}\rceil+1)$ on the CA B in this following way (cf. Fig. 8):

(1) at time $2n-1+k(\lceil\sqrt{n}\rceil+1)$ providing the step k of the CA A is computed the new states are propagated at maximal speed to the left and to the right;

(2) at the next step $2n+k(\lceil\sqrt{n}\rceil+1)$ in order to reverse each segment the digits are sent at maximal speed to the right, bounce on the rightmost cell of the segment and return at maximal speed to the left, in parallel a synchronization process is initiated from both ends of each segment;

(3) in this way at time $2n+k(\lceil\sqrt{n}\rceil+1)+\lceil\sqrt{n}\rceil-1$ the synchronization occurs. Moreover, the ith cell of the jth segment receives the ith states of the $(j-1)$th and $(j+1)$th segments and the $(\lceil\sqrt{n}\rceil-i)$th state of the jth segment which correspond to the states of the cells $(i,j-1),(i,j),(i,j+1)$ (resp. $(\lceil\sqrt{n}\rceil-i,j-1),(\lceil\sqrt{n}\rceil-i,j),(\lceil\sqrt{n}\rceil-i,j+1)$) of the CA A if $j+k$ is even (resp. odd).

Thus at the next step $2n-1+(k+1)(\lceil\sqrt{n}\rceil+1)$ we know all the required neighborhood to compute the new states corresponding to the step $k+1$ of the CA A.

In fact the computation of the last $n-(\lceil\sqrt{n}\rceil-1)^2$ cells are performed on the segment $\lceil\sqrt{n}\rceil-1$.

5. Comparison with Turing machine

Here as two-dimensional CAs are restricted to bounded computation, two-dimensional CAs will have the same power than Turing machines bounded in space n. Now we describe direct simulations.

5.1. Simulation of a Turing machine working in space n by a two-dimensional CA

Proposition 5. *If L is recognized by a Turing machine which works in time $f(n)$ and space n then L is recognized in time $\lceil\sqrt{n}\rceil+f(n)$ by a two-dimensional CA.*

Proof. The construction of the two-dimensional CA is based on the usual techniques [9] where the simulated heads cells remain fixed on the site $(1,1)$ and the simulated tapes are shifted. In order that the simulation is performed in real time, the simulated

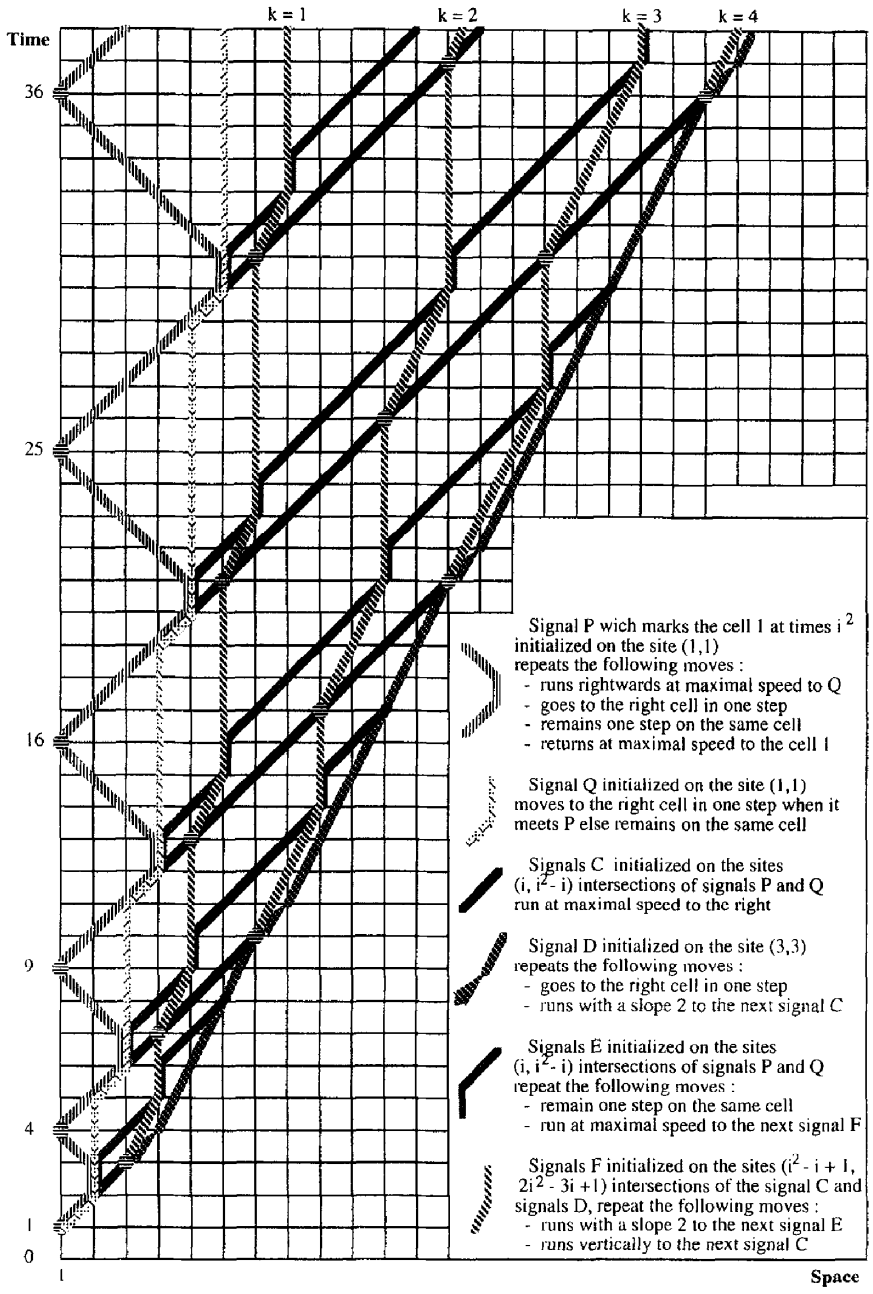

Fig. 7. Construction of the sites $(1+ik,(i-1)^2+ik)$ intersections of the signals C and F.

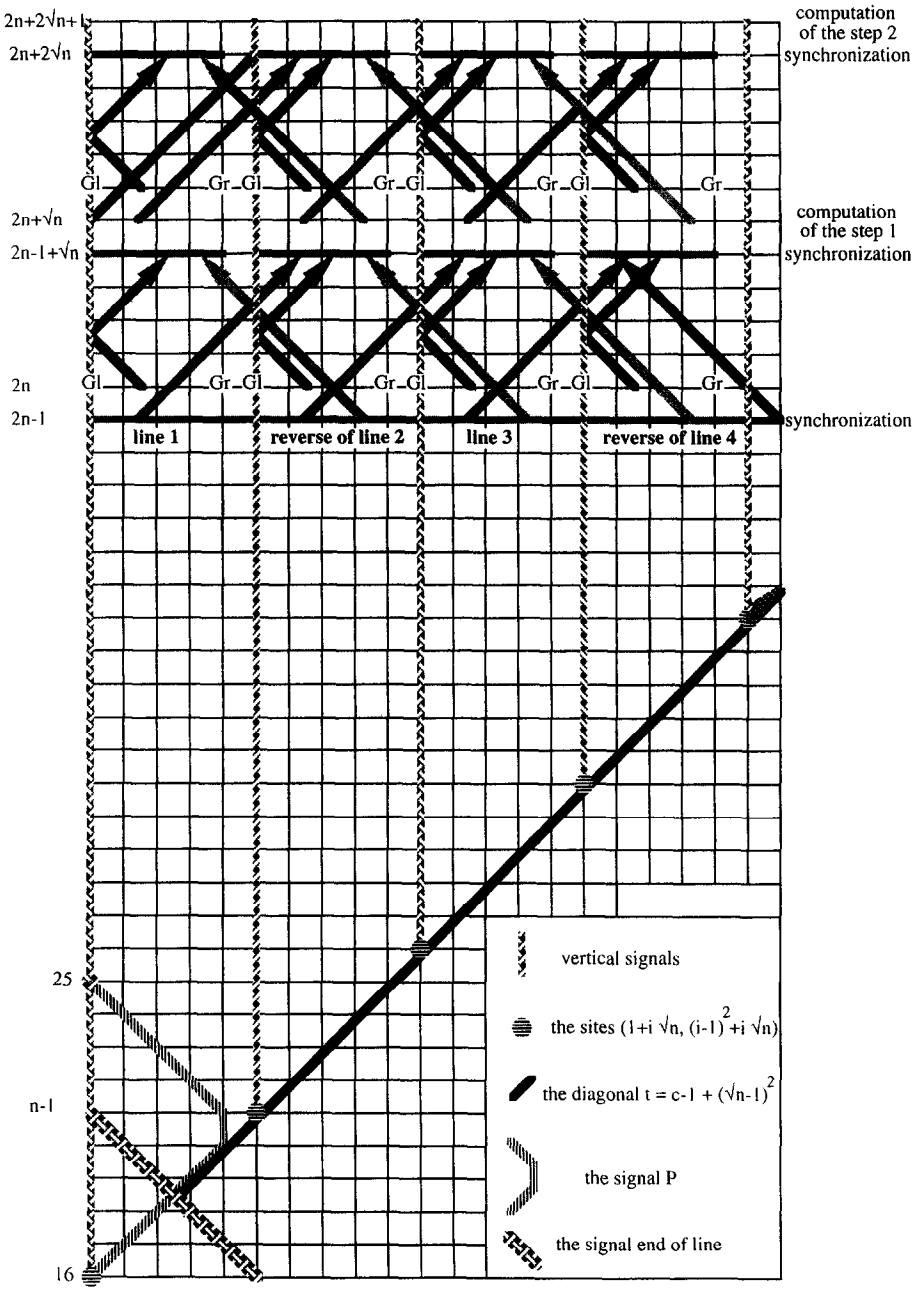
2n+2√n+1
2n+2√n
2n+√n
2n-1+√n
2n
2n-1
25
n-1
16
Gl
Gr
computation of the step 2
synchronization
computation of the step 1
synchronization
synchronization
line 1
reverse of line 2
line 3
reverse of line 4
vertical signals
the sites (1+i √n, (i-1)²+i √n)
the diagonal t = c-1 + (√n-1)²
the signal P
the signal end of line

Fig. 8.

squares must be grouped. So an initial phase of $\lceil\sqrt{n}\rceil$ steps is required to group the columns by two to the left and also to exhibit, as in Section 4.1, the trace of the simulated tapes. As the Turing machine works in space n, this trace is contained in the bounded area.

Proposition 5 can also be obtained as a consequence of Proposition 3.

5.2. Simulation of a two-dimensional CA by a Turing machine

Proposition 6. *Let L be a one-dimensional language recognized in time $f(n)$ by a two-dimensional CA then L is recognized in time $O(nf(n))$ and space n by a Turing machine with two tapes.*

Proof. We construct a Turing machine with two tapes, the second tape with two tracks. On the CA an input word of length n is supplied in a snakelike order as a picture of size $(\lceil\sqrt{n}\rceil, \lceil\sqrt{n}\rceil)$. So the initial phase of the simulation consists in dividing the input word of length n into $\lceil\sqrt{n}\rceil$ blocks of length $\lceil\sqrt{n}\rceil$ so that the ith block corresponds to the ith line if i is odd, the reverse of the ith line if i is even. A first scan of the input permits to compute $\lceil\sqrt{n}\rceil$ on the second tape. A second scan of the input is required to mark each $i.\ \lceil\sqrt{n}\rceil$th letters (for $i=1,\ldots,\lceil\sqrt{n}\rceil$) of the input by $*$.
So the initial phase takes $O(n)$ steps.

Then the simulation of one step of the CA goes as follows. See Fig. 9. On the first tape the head scans the input one square by step. On the second tape the head scans $\lceil\sqrt{n}\rceil$ squares successively from left to right and from right to left. During the left-to-right sweeps it records on the first track the odd blocks (red on the first tape) and during the right-to-left sweeps it records on the second track the even blocks (red on the first tape).

By this way at the end of the scan of the ith block on the first tape, we have the ith line on the upper (resp. lower) track, the $(i-1)$th line on the lower (resp. upper) track and the head points to the rightmost (resp. leftmost) square if i is even (resp. odd). So simultaneously the heads will point to the $(j+1)$th elements of the $(i-1)$th, ith, $(i+1)$th lines; and if in the state the contents of the squares scanned during the two last steps (the $(j-1)$th and the jth elements of the $(i-1)$th, ith, $(i+1)$th lines) are recorded we know all the relevant information to compute the jth element of the ith line at the next step of the CA.

Thus simulating one step of the CA requires $n+\lceil\sqrt{n}\rceil+1$ steps on the TM and the lines are shifted $\lceil\sqrt{n}\rceil+1$ squares rightward. Now it is the odd line which are written in reverse. So the simulation of the next step is made in the same way but in reverse.

6. Real-time recognition

In this section we will exploit methods, developed before in one-dimensional case [2, 10] which used equivalence classes and counting arguments to show that some languages are not real time recognizable.

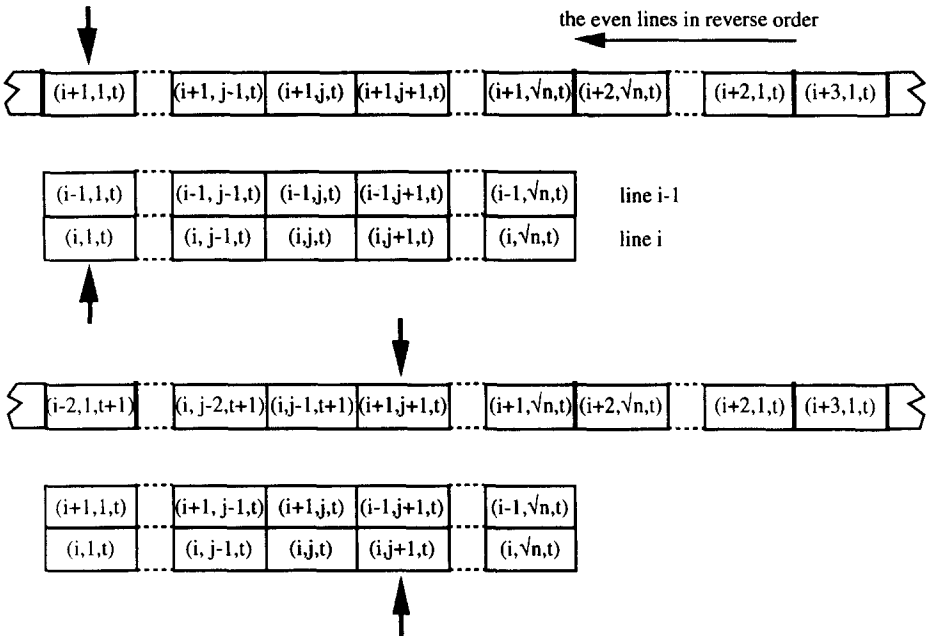

Fig. 9. Simulation of the line i.

6.1. *The Moore neighborhood case*

A characteristic of 2-CAs in real time with Moore neighborhood is that on picture of size (n,n) the input element of the cell (n,n) has only one way (the diagonal) to reach the distinguished cell(1,1) and then this element interacts with only $O(n)$ elements among the n^2 elements of the input array. This feature allow us to express a relation of equivalence. Then we will present a language which will be shown by counting arguments not to be a real time language for CAs with Moore neighborhood. Finally we will define a CA with Von Neumann neighborhood able to recognize it in real time.

Notation. See Fig. 10. Let M be a 2-CA with Moore neighborhood with (1,1) as distinguished cell. We consider the behavior of the CA working in real time on input pictures of size (m,n). We will use the following notations. For any set $U \subset Z^2$ we denote by $\mathrm{Neig}(U)$ the Moore neighborhood of U and $\mathrm{Neig}^t(U)$ its tth iterate. Let A be the set $\{(x,y): 1 \leqslant x \leqslant m, 1 \leqslant y \leqslant n\}$ and $E_t = \mathrm{Neig}^{\max(m,n)-1-t}(\{(1,1)\}) \cap A$ denote the set of the cells at step t which can have an effect on the result of the computation, i.e. which can affect the cell (1,1) at time $\max(m,n)-1$.

Let $R_t(U) = \mathrm{Neig}^t(U) \cap E_t$ represent the set of cells which contain at time t the relevant information regarding the input bits supplied at initial time in U.

A portion of picture p on a set U consists in the elements $p(i,j)$ such that $(i,j) \in U$.

We denote by $p \oplus c$ the picture resulting of the concatenation of a portion of picture p on A/U and a portion of picture c on U.

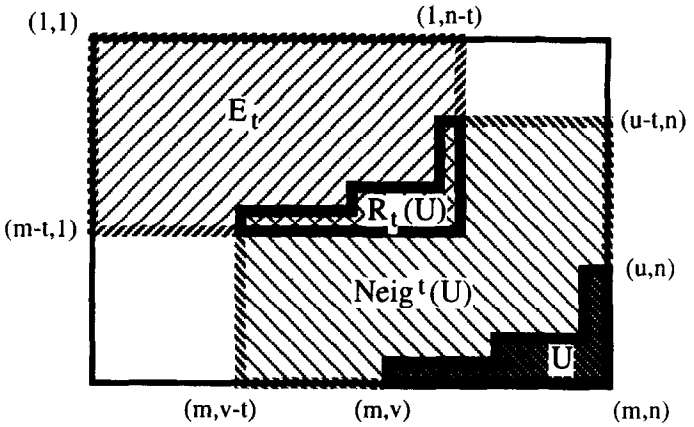

Fig. 10.

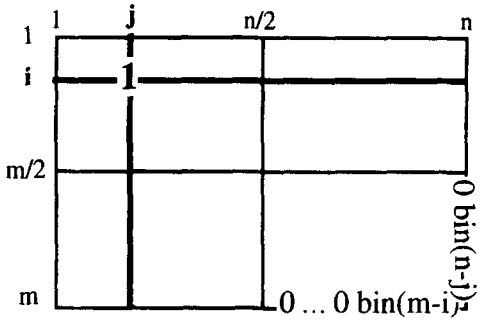

Fig. 11. The language L.

The relation of equivalence. Note that by definition of $R_t(U)$ the states of $\mathrm{Neig}(R_{t+1}(U))/R_t(U)$ at time $t=0,\ldots,\max(m,n)-1$ are independent of the input bits supplied in U at time 0. So we could define the following relation of equivalence: two portions of picture p, p' on A/U are *U-equivalent* if the two states sequences of the cells $\mathrm{Neig}(R_{t+1}(U))/R_t(U)$ at time $t=0,\ldots,\max(m,n)-2$ during the evolution of the CA on the inputs p and p' are equal.

Fact 1. *If p and p' are U-equivalent then for all portions of picture c on U, $p\oplus c$ is accepted by the CA if and only if $p'\oplus c$ is accepted by the CA.*

Proof. As the states of R_t at time t are function of the states of $\mathrm{Neig}(R_t(U))/R_{t-1}(U)$ and the states of $R_{t-1}(U)$ at time $t-1$, if p and p' are U-equivalent then for all portions of picture c on U the sequences of states of $R_t(U)$ at time $t=1,\ldots,\max(m,n)-1$ generated by the CA on the inputs $p\oplus c$ and $p'\oplus c$ are equal. In particular, as $R_{\max(m,n)-1}(U)=\{(1,1)\}$ if p and p' are U-equivalent then for all portions of picture c on U we have $p\oplus c$ is accepted by the CA if and only if $p'\oplus c$ is accepted by the CA.

Let $L=\{p\in\{0,1\}^{**}$: p is of size (m,n) with $m\geqslant 2(\log(n)+2)$ and $n\geqslant 2(\log(m)+2)$ and it exists i,j such that $1\leqslant i\leqslant m/2, 1\leqslant j\leqslant n/2, p(i,j)=1$, $p(m,\lfloor\frac{n}{2}\rfloor+1)=\cdots=p(m,n-\lfloor\log(m-i)\rfloor-2)=0$, $p(m,n-\lfloor\log(m-i)\rfloor-1)\cdots p(m,n-1)$ is the binary notation of $m-i$, $p(\lfloor\frac{m}{2}\rfloor+1,n)=\cdots=p(m-\lfloor\log(n-j)\rfloor-2,n)=0$, $p(m-\lfloor\log(n-j)\rfloor-1,n)\cdots p(m-1,n)$ is the binary notation of $n-j\}$. See Fig. 11.

Proposition 7. *L is not recognizable in real time by 2-CAs with Moore neighborhood.*

Proof. We consider pictures A of size (n,n) and the set $U=\{(x,n)\colon x=n-\lfloor\log(n-1)\rfloor-1,\ldots,n-1\}\cup\{(n,y)\colon y=n-\lfloor\log(n-1)\rfloor-1,\ldots,n-1\}$. Note that the cardinality of $\mathrm{Neig}(R_t(U))/R_{t-1}(U)$ is lower than $4\lfloor\log(n-1)\rfloor+8$. Thus the cardinality of $\bigcup_{t=0}^{n-2}\mathrm{Neig}(R_t(U))/R_{t-1}(U)$ is lower than $(n-1)(4\lfloor\log(n-1)\rfloor+8)$. So if L is recognized by a Moore CA in real time whose the number of states is s then there is at most $s^{(n-1)(4\log(n-1)+8)}=2^{O(n\log(n))}U$ -equivalence classes for the portions of picture on A/U.

At each subset E of $\{(x,y)\colon 1\leqslant x,y\leqslant n/2\}$ we associate the portion of picture p_E on A/U defined by $p_E(x,y)=0$ if $x>n/2$ or $y>n/2$ and for $1\leqslant x,y\leqslant n/2$ $p_E(x,y)=1$ if $(x,y)\in E$, else $p_E(x,y)=0$. Observe that for all distinct subsets E,F of $\{(x,y)\colon 1\leqslant x, y\leqslant n/2\}$ there exists integers x,y such that (x,y) belongs to E/F or F/E. Now consider the portion of picture c on U such that $c(n,n-\lfloor\log(n-1)\rfloor-1)=\cdots=c(n,n-\lfloor\log(n-i)\rfloor-2)=0$, $c(n,n-\lfloor\log(n-i)\rfloor-1)\ldots c(n,n-1)$ is the binary notation of $n-i$, $c(n-\lfloor\log(n-1)\rfloor-1,n)=\cdots=c(n-\lfloor\log(n-j)\rfloor-2,n)=0$, $c(n-\lfloor\log(n-j)\rfloor-1,n)\ldots c(n-1,n)$ is the binary notation of $n-j$. Thus $p_E\oplus c$ belongs to L if and only if $p_F\oplus c$ does not belong to L. In other words p_E and p_F are not equivalent. To conclude note that the number of subsets of $\{(x,y)\colon 1\leqslant x,y\leqslant n/2\}$ is $2^{n^2/4}$. Hence it is of order greater than $2^{O(n\log(n))}$.

Proposition 8. *L is recognizable in real time by a 2-CA with Von Neumann neighborhood.*

Proof. See Fig. 12. From $p(m,\lfloor\frac{n}{2}\rfloor+1),\ldots,p(m,n-1)=0\ldots0\ \mathrm{bin}(m-i)$, if $1\leqslant i\leqslant m/2$ we have to select the line i. For that purpose we compute $\mathrm{bin}(m-k)$ on the right of the kth line for all $k=m-1,\ldots,1$ in the usual way. We consider that initially all cells are in a quiescent state β. The process begins on the cell $(m-1,n-1)$ which can be distinguished at time 2 and enters in state 1. Then the CA evolves according this rules:

$$\begin{matrix} & n & & \\ w & c & e & \to c= \\ & s & & \end{matrix}\begin{cases} 1 & \text{if } s=e=1 \text{ or } (e=0' \text{ and } s=0,\ \beta),\\ 0 & \text{if } e=1 \text{ and } s=0,\ 0',\\ 0' & \text{if } s=1 \text{ and } e=0',\ \beta. \end{cases}$$

So the tth bit of $\mathrm{bin}(m-k)$ is computed on the cell $(k,n-t)$ at time $m-k+t$. Simultaneously from the step 1 the lowest line is sent upward, so the element $p(m,t)$

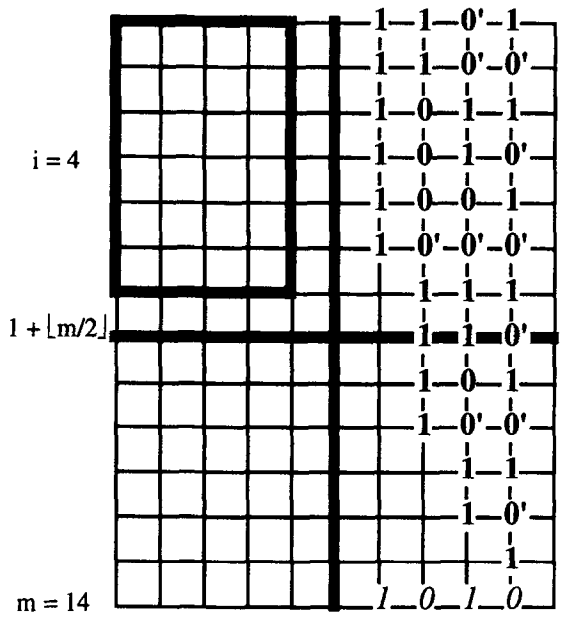

Fig. 12.

is known by the cell (k,t) at time $m-k+1$. In addition we can distinguish the line $1+\lfloor m/2\rfloor$ and the column $1+\lfloor n/2\rfloor$ by sending signals from both ends of each line and each column. Thus a comparison process can start on each line k with $k \leqslant n/2$ at time $m-k+1$ from the cell $(k,n-1)$ and propagates to the right until reaching the column $1+\lfloor n/2\rfloor$. The comparison succeeds on the line u. And from the site $(i,1+\lfloor n/2\rfloor)$ a signal of selection is sent to the right. Inverting the roles of the lines and the columns the same process is achieved in order to select the column j. In this way the cell (i,j) is selected at time $m-i+n-j$. To sent its content to the cell $(1,1)$ requires $i+j-2$ steps, hence this content reaches the cell $(1,1)$ at time $m+n-2$.

6.2. *The Von Neumann case*

Here we will restrict to a particular case where the distinguished cell is the cell $(m,1)$ for pictures of size $(2m-1,n)$. We consider the behavior of a 2-CA with Von Neumann neighborhood working in real time on input pictures of size $(2m-1,n)$. We need the following notations.

Let t be an integer less than n. We denote by A the set $\{(x,y)\colon 1 \leqslant x \leqslant 2m-1, 1 \leqslant y \leqslant n\}$, by U the set of t cells $\{(1,y)\colon n-t<y \leqslant n\}$, by V the set of t cells $\{(2m-1,y)\colon n-t<y \leqslant n\}$. We focus on the step $m-2$. $E_{m-2}=\mathrm{Neig}^n(\{(m,1)\})=\{(x,y)\colon 1 \leqslant y, x-y \geqslant m-n-1, x+y \leqslant m+n+1\}$ denotes the set of cells at step $m-2$ which can have an effect on the result of the computation (i.e. the cell $(m,1)$ at time $m+n-2$).

$\mathrm{Neig}^{m-2}(U)=\{(x,y)\colon 1\leqslant x\leqslant m-1, 1\leqslant y\leqslant n, x-y\leqslant m-n+t-2\}$ (resp. $\mathrm{Neig}^{m-2}(V)=\{(x,y)\colon m+1\leqslant x\leqslant 2m-1, 1\leqslant y\leqslant n, x+y\geqslant m+n-t+2\}$) represents the set of cells at time $m-2$ depending on the input bits supplied at time 0 in U (resp. V).

Note that at time $m-2$, $E_{m-2}/\mathrm{Neig}^{m-2}(U\cup V)$ is independent of the input bits supplied at initial time in U and V, $\mathrm{Neig}^{m-2}(U)$ is independent of V and $\mathrm{Neig}^{m-2}(V)$ of U. So we could define the following relation of equivalence: two portions of picture p, p' on $A/(U\cup V)$ are (U,V)-*equivalent* if

(i) the states at time $m-2$ of $E_{m-2}/\mathrm{Neig}^{m-2}(U\cup V)$ of the CA on the inputs p and p' are equal;

(ii) for all portions of picture u on U, the states at time $m-2$ of $E_{m-2}\cap\mathrm{Neig}^{m-2}(U)$ of the CA on the inputs $p\oplus u$ and $p'\oplus u$ are equal;

(iii) for all portions of picture v on V, the states at time $m-2$ of $E_{m-2}\cap\mathrm{Neig}^{m-2}(V)$ of the CA on the inputs $p\oplus v$ and $p'\oplus v$ are equal.

Fact 2. *The number of (U,V)-equivalence classes of portions of pictures on $A/(U\cup V)$ is bounded by $2^{O(n2^t)}$.*

Proof. Let s be the number of states of the CA. (i) As the cardinality of $E_{m-2}/\mathrm{Neig}^{m-2}(U\cup V)$ is $n+(n-t)(n-t+1)$ there is at most $s^{n+(n-t)(n-t+1)}$ distinguishable pictures on $E_{m-2}/\mathrm{Neig}^{m-2}(U\cup V)$. (ii) Note that $\mathrm{Neig}^{m-2}(U)\cap E_{m-2}$ is composed first for $i=1,\ldots,t$ of the $(n-t+i/2)$ sites $\{(y+m-n+t-1, y)\colon 1\leqslant y\leqslant n-t+(i+1)/2\}$ which depends on the bits supplied at initial time on the i cells $(1,n-t+1),\ldots,(1,n-t+i)$ but not on the $t-i$ cells $(1,n-t+i+1),\ldots,(1,n)$, second for $2\leqslant i\leqslant j\leqslant t$ and $i+j$ even of the sites $(m-1+(j-i)/2, n-t+(i+j)/2)$ which depend exactly on the bits supplied at initial time on the $j-i+1$ cells $(1,n-t+i),\ldots,(1,n-t+j)$. So the number of distinguishable pictures on $\mathrm{Neig}^{m-2}(U)\cap E_{m-2}$ at time $m-2$ is at most $\prod_{1\leqslant i\leqslant t} s^{(n-t+\lfloor i/2\rfloor)2^i}\cdot\prod_{\substack{2\leqslant i\leqslant j\leqslant t\\ i+j \text{ even}}} s^{1+j-i}=2^{c(n2^t+t^3)}$. (iii) In the same way the number of distinguishable pictures on $\mathrm{Neig}^{m-2}(U)\cap E_{m-2}$ at time $m-2$ is at most $2^{c(n2^t+t^3)}$. Thus the number of (U,V)-equivalence classes is at most $2^{c(n+(n-t)(n-t+1)}\cdot 2^{c(n2^t+t^3)}\cdot 2^{c(n2^t+t^3)}$ i.e. $2^{O(n2^t)}$.

We present now a language not real-time recognizable.

Let $a=\lceil\log n\rceil$, $b=\lceil\log 2m\rceil$ and $t=\lceil(a+b)/2\rceil$.

Let $L=\{p\in\{0,1\}^{**}\colon p$ is of size $(2m-1,n)$ and there exist i,j
such that $1\leqslant i\leqslant 2m-1, 1\leqslant j\leqslant n$,
$p(i,j)=1, p(1,n-t+1)\ldots p(1,n-t+a)$ is the binary notation of j,
$p(1,n-t+a+1)\ldots p(1,n)p(2m-1,n-t+1)\ldots p(2m-1,n)$
is the binary notation of $i\}$.

Proposition 9. *L is not recognizable in real time by a 2-CA with Von Neumann neighborhood whose cell $(m,1)$ is the distinguished cell.*

Proof. At each subset E of $\{(x,y)\colon 1<x<2m-1, 1\leqslant y\leqslant n\}$ we associate the portion of picture p_E on $A/(U\cup V)$ defined by $p_E(x,y)=1$ if $(x,y)\in E$ else $p_E(x,y)=0$. As $a=\lceil\log n\rceil$ and $b=\lceil\log 2m\rceil$ all lines and columns could be encoded. Hence for all distinct subsets E and F of $\{(x,y)\colon 1<x<2m-1, 1\leqslant y\leqslant n\}$, p_E and p_F are not (U,V)-equivalent. Note that as $t=(a+b)/2$ the number of (U,V)-equivalence classes for portions of pictures on $A/(U\cup V)$ is bounded by $2^{O(n2^{(a+b)/2})}$. In other part, the number of subsets of $\{(x,y)\colon 1\leqslant x\leqslant n, 1<y<2m-1\}$ is $2^{n(2m-3)}$. Hence, provided $m=O(2^b)$ is of greater order than $n=O(2^a)$, $2^{n(2m-3)}$ is of order greater than $2^{O(n2^{(a+b)/2})}$.

References

[1] A.C. Achilles, M. Kutrib, T. Worsch, On relations between arrays of processing elements of different dimensionality, Parcella 96, Akademie Verlag, Berlin, 1996, pp. 13–20.

[2] S.N. Cole, Real-time computation by n-dimensional iterative arrays of finite-state machine, IEEE Trans. Comput. C-18 (1969) 349–365.

[3] O.H. Ibarra, S.M. Kim, S. Moran, Sequential machine characterization of trellis and cellular automata and application, SIAM J. Comput. 14 (1985) 426–447.

[4] O.H. Ibarra, M.A. Palis, Two-dimensional iterative arrays: characterizations and applications, Theoret. Comput. Sci. 56 (1988) 47–86.

[5] K. Inoue, A. Nakamura, Some properties of two-dimensional on-line tessalation acceptors, Inform. Sci. 13 (1977) 95–121.

[6] J. Kari, Reversibility of 2D cellular automata is undecidable, Physica D 45 (1990) 379–385.

[7] J. Mazoyer, N. Reimen, A linear speed-up theorem for cellular automata, Theoret. Comput. Sci. 101 (1992) 59–98.

[8] Zs. Roka, Simulations between cellular automata on Cayley graphs, Theoret. Comput. Sci. (1999), to appear.

[9] A.R. Smith, Real-time language recognition by one-dimensional cellular automata, J. ACM 6 (1972) 233–253.

[10] V. Terrier, Language not recognizable in real time by one-way celular automata, Theoret. Comput. Sci. 156 (1996) 281–287.